



RRRRRRRR	MM	MM	SSSSSSSS	000000	PPPPPPPP	UU	UU	TTTTTTTT
RRRRRRRR	MM	MM	SSSSSSSS	000000	PPPPPPPP	UU	UU	TTTTTTTT
RR RR	RR	MMMM	MMMM	SS	00 00	PP PP	UU	UU TT
RR RR	RR	MMMM	MMMM	SS	00 00	PP PP	UU	UU TT
RR RR	RR	MM MM	MM MM	SS	00 0000	PP PP	UU	UU TT
RR RR	RR	MM MM	MM MM	SS	00 0000	PP PP	UU	UU TT
RRRRRRRR	MM	MM	SSSSSS	00 00	PPPPPPPP	UU	UU	TT
RRRRRRRR	MM	MM	SSSSSS	00 00	PPPPPPPP	UU	UU	TT
RR RR	RR	MM MM	MM MM	SS	0000 00	PP	UU	UU TT
RR RR	RR	MM MM	MM MM	SS	0000 00	PP	UU	UU TT
RR RR	RR	MM MM	MM MM	SS	00 00	PP	UU	UU TT
RR RR	RR	MM MM	MM MM	SS	00 00	PP	UU	UU TT
RR RR	RR	MM MM	MM MM	SSSSSS	000000	PP	UUUUUUUUUU	TT
RR RR	RR	MM MM	MM MM	SSSSSS	000000	PP	UUUUUUUUUU	TT

....

LL		SSSSSSSS
LL		SSSSSSSS
LL		SS
LL		SS
LL		SS
LL		SSSSSS
LL		SSSSSS
LL		SS
LL		SS
LL		SS
LLLLLLLL		SSSSSSSS
LLLLLLLL		SSSSSSSS

(2) 71 DECLARATIONS  
(3) 94 RMSSPUT - COMMON \$PUT SETUP AND DISPATCH ROUTINE

```
0000 1      $BEGIN RMSOPUT,000,RM$RMS,<DISPATCH FOR PUT OPERATION>,<NOWRT,QUAD>
0000 2
0000 3
0000 4      ****
0000 5      *
0000 6      * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7      * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8      * ALL RIGHTS RESERVED.
0000 9      *
0000 10     * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11     * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12     * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13     * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14     * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15     * TRANSFERRED.
0000 16     *
0000 17     * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18     * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19     * CORPORATION.
0000 20     *
0000 21     * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22     * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23     *
0000 24     *
0000 25      ****
0000 26
0000 27      ++
0000 28      Facility: rms32
0000 29
0000 30      Abstract:
0000 31      this routine is the highest level control
0000 32      routine to perform the $put function.
0000 33
0000 34
0000 35
0000 36      Environment:
0000 37      star processor running starlet exec.
0000 38
0000 39      Author: L F Laverdure,           creation date: 3-FEB-1977
0000 40
0000 41      Modified By:
0000 42
0000 43      V03-007 DAS0001      David Solomon      14-Apr-1984
0000 44      Fix truncation error in CASE to RM$PUT2.
0000 45
0000 46      V03-006 JWT0141      Jim Teague      11-Nov-1983
0000 47      Change IFBSV_RUM to IFBSV_ONLY_RU
0000 48
0000 49      V03-005 KPL0003      Peter Lieberwirth  26-Jul-1983
0000 50      If AT jnling, tell RJR we're a PUT.
0000 51
0000 52      V03-004 KPL0002      Peter Lieberwirth  24-Jul-1983
0000 53      If AT jnling, get RAB data describing user's request.
0000 54
0000 55      V03-003 KPL0001      Peter Lieberwirth  20-Jun-1983
0000 56      Change some JNLFLG references to JNLFLG2.
0000 57
```

## DISPATCH FOR PUT OPERATION

F 10

16-SEP-1984 01:27:12 VAX/VMS Macro V04-00  
5-SEP-1984 16:25:18 [RMS.SRC]RMSOPUT.MAR;1Page 2  
(1)

0000	58	:	V03-002 JWH0153	Jeffrey W. Horn	8-Dec-1982
0000	59	:		Don't allow \$PUT if not in recovery unit and RU only	
0000	60	:		specified for file.	
0000	61	:			
0000	62	:	V03-001 KBT0189	Keith B. Thompson	23-Aug-1982
0000	63	:		Reorganize psects	
0000	64	:			
0000	65	:	V02-005 REFORMAT	Maria del C. Nasr	24-Jul-1980
0000	66	:			
0000	67	:	--		
0000	68	:			
0000	69				

```
0000 71 .SBTTL DECLARATIONS
0000 72
0000 73 :
0000 74 : Include Files:
0000 75 :
0000 76
0000 77 :
0000 78 : Macros:
0000 79 :
0000 80
0000 81     $IFBDEF
0000 82     $RMSDEF
0000 83     $RJRDEF
0000 84
0000 85 :
0000 86 : Equated Symbols:
0000 87 :
0000 88
0000 89 :
0000 90 : Own Storage:
0000 91 :
0000 92
```

0000 94 .SBTTL RMSSPUT - COMMON \$PUT SETUP AND DISPATCH ROUTINE  
0000 95  
0000 96 :++  
0000 97 : RMSSPUT - This routine performs common RAB function setup followed  
0000 98 : by dispatch to organization-dependent \$PUT code.  
0000 99  
0000 100 : Calling sequence:  
0000 101  
0000 102 : entered from exec as a result of user's calling sys\$put  
0000 103 : (e.g., by using the \$put macro)  
0000 104  
0000 105 : Input Parameters:  
0000 106 :  
0000 107 : ap user's argument list addr  
0000 108 :  
0000 109 : Implicit Inputs:  
0000 110 :  
0000 111 : the contents of the rab and related irab and ifab.  
0000 112 :  
0000 113 : Output Parameters:  
0000 114 :  
0000 115 : r1 destroyed  
0000 116 : r0 status code  
0000 117 :  
0000 118 : Implicit Outputs:  
0000 119 :  
0000 120 : various fields of the rab are filled in to reflect  
0000 121 : the status of the \$put operation. (see rms functional  
0000 122 : spec for a complete list.)  
0000 123 :  
0000 124 : the irab is similarly updated.  
0000 125 :  
0000 126 : a completion ast is queued if specified in the user arglist.  
0000 127 :  
0000 128 : Completion Codes:  
0000 129 :  
0000 130 : standard rms (see functional spec for list).  
0000 131 :  
0000 132 : Side Effects:  
0000 133 :  
0000 134 : none  
0000 135 :  
0000 136 :--  
0000 137  
0000 138 :\$ENTRY RMSSPUT  
0000 139 :\$TSTPT PUT  
0006 140 :\$RABSET FAC=IFB\$V\_PUT,CFLG=1 ; do common setup  
000A 141  
000A 142 :  
000A 143 : Returns to user on error  
000A 144 :  
000A 145 :  
OE 00A0 CA 00 E1 000A 146 :BBC #IFB\$V\_ONLY RU,IFB\$B\_JNLFLG(R10),10\$ ; branch if not RU only  
0B 00A2 CA 02 E0 0010 147 :BBS #IFB\$V\_RUP,IFB\$B\_JNLFLG2(R10),10\$ ; branch if in RU  
FFE2' 31 001B 148 :RMSERR NRU  
FFE2' 31 001B 149 :BRW RMSEXRMS  
FFE2' 31 001E 150

001E 151 10\$:  
001E 152  
001E 153 :  
001E 154 : If AT journaling, get some information from RAB.  
001E 155 :  
09 00A0 CA 04 E1 001E 156 BBC #IFBSV\_AT,IFBSB\_JNLFLG(R10),20\$ ; skip if not AT jnling  
51 13 D0 0024 157 MOVL #RJRS\_PUT,R1 ; input to AT COM\_RAB  
00000000'EF 16 0027 158 JSB RMSAT\_COM\_RAB ; get RAB data into RJR  
002D 159 20\$:  
002D 160  
002D 161 :  
002D 162 : Dispatch to org-dependent code  
002D 163 : Sequential, Relative, indexed routines  
002D 164 :  
002D 165  
002D 166 CASE TYPE=B, SRC=IFBSB\_ORGCASE(R10),-  
002D 167 DISPLIST=<RMSSPUT1,RM\_PUT2\_BR,RMSSPUT3>  
00000008 0038 168 .IF NE \$\$RMSTEST&\$\$RMS\_TBUGCHR  
FFC5' 31 0038 169 BRW RM\$ERRORG  
003B 170 .ENDC  
00000000'EF 17 003B 171 RM\_PUT2\_BR:  
0041 172 JMP RMSSPUT2  
0041 173  
0041 174 .END

```

$$.PSECT EP          = 00000000
$$.RMSTEST          = 0000001A
$$.RMS_PBUGCHK      = 00000010
$$.RMS_TBUGCHK      = 00000008
$$.RMS_UMODE         = 00000004
IFBSB_JNLFLG        = 000000A0
IFBSB_JNLFLG2       = 000000A2
IFBSB_ORGCASE       = 00000023
IFBSV_AT            = 00000004
IFBSV_ONLY_RU       = 00000000
IFBSV_PUT            = 00000000
IFBSV_RUP            = 00000002
PIOSA_TRACE          = *****
RJRS_PUT             = 00000013
RMSAT_COM_RAB        = *****
RMSErrorOrg          = *****
RMSEXRMS             = *****
RMSPUT1              = *****
RMSPUT2              = *****
RMSPUT3              = *****
RMSRSET              = *****
RMSSPUT              = FFFFFFFE RG 01
RMSS_NRU              = 000187FC
RM_PUT2_BR            = 0000003B R 01
TPTSL_PUT             = ***** X 01

```

-----+  
! Psect synopsis !  
-----+

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
RMSRMS	00000041 ( 65.)	01 ( 1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC QUAD
SABSS	00000000 ( 0.)	02 ( 2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

-----+  
! Performance indicators !  
-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.52
Command processing	111	00:00:00.72	00:00:05.29
Pass 1	227	00:00:05.38	00:00:14.83
Symbol table sort	0	00:00:00.72	00:00:00.84
Pass 2	45	00:00:01.01	00:00:02.01
Symbol table output	4	00:00:00.05	00:00:00.33
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	420	00:00:07.99	00:00:23.93

The working set limit was 1200 pages.  
 28972 bytes (57 pages) of virtual memory were used to buffer the intermediate code.  
 There were 30 pages of symbol table space allocated to hold 584 non-local and 4 local symbols.  
 174 source lines were read in Pass 1, producing 13 object records in Pass 2.  
 18 pages of virtual memory were used to define 17 macros.

! Macro library statistics !

Macro library name

\$255\$DUA28:[RMS.OBJ]RMS.MLB;1  
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1  
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2  
TOTALS (all libraries)

Macros defined

9  
1  
3  
13

696 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RMSOPUT/OBJ=OBJ\$:RMSOPUT MSRC\$:RMSOPUT/UPDATE=(ENH\$:RMSOPUT)+EXECML\$/LIB+LIB\$:RMS/LIB

0330 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

RMS0PUT  
LIS

RMS0MAGTA  
LIS

RMS0RNDWN  
LIS

RMS0REWIN  
LIS

RMS0SETDD  
LIS

RMS0LSICH  
LIS

RMS0MISC  
LIS

RMS0OPEN  
LIS

RMS0PARSE  
LIS

RMS0RUHND  
LIS

RMS0MODFY  
LIS

RMS0RENAME  
LIS

RMS0SOFP  
LIS

